Text Simplification for Children

Jan De Belder
Katholieke Universiteit Leuven
Department of Computer Science
Celestijnenlaan 200A
B-3001 Heverlee, Belgium
jan.debelder@cs.kuleuven.be

Marie-Francine Moens
Katholieke Universiteit Leuven
Department of Computer Science
Celestijnenlaan 200A
B-3001 Heverlee, Belgium
sien.moens@cs.kuleuven.be

ABSTRACT

The goal in this paper is to automatically transform text into a simpler text, so that it is easier to understand by children. We perform syntactic simplification, i.e. the splitting of sentences, and lexical simplification, i.e. replacing difficult words with easier synonyms. We test the performance of this approach for each component separately on a per sentence basis, and globally with the automatic construction of simplified news articles and encyclopedia articles. By including information from a language model in the lexical simplification step, we obtain better results over a baseline method. The syntactic simplification shows that some phenomena are difficult to recognize by a parser, and that errors are often introduced. Although the reading difficulty goes down, it still doesn't reach the required level for young children.

Keywords

text simplification, readability

1. INTRODUCTION

The Internet contains a wealth of information, but only a small fraction of that information is suited for the reading level of children. Especially in the last decade, a lot of research has been put into automatically assigning a measure of readability to text, and retrieving documents that are suited for a predetermined reading level. This paper addresses a related issue, that arises when a document with the right reading level can't be found: rewrite the text so that it does become suited, according to an external readability measure. We introduce a method that takes complicated text as input, and generates a text that is simpler and easier to understand for children.

Text simplification may serve many purposes, and has been researched with very different objectives in mind. Originally, the purpose was to break down long sentences in order to improve the accuracy of parsers [4, 26]. Text simplification was also used to automatically make text more understandable by aphasic readers [3], or readers with low literacy skills [2]. Yet another application is the simplification of text as a preprocessing step for other NLP tasks, such as Relation Extraction [16], Semantic Role Labeling [27] and Machine Translation [21].

The goal of most research on text simplification is to make the text as simple as possible. Only [23] and [2] first train a classifier that decides whether or not a sentence is too difficult, and if it is the case then a rule based system is applied to attempt to simplify the sentence. The problem with training a classifier is that annotated training data is needed, and even then the decisions are made on the level of individual sentences, not on the level of the entire document. The problem with simplifying as much as possible is that the text might become too easy: we want the text to fit the reading level of a child as good as possible, rather than making it overly simple.

By casting the problem as an Integer Linear Programming (ILP) problem, we can find a global solution (i.e. choice of simplifications) so that the entire text satisfies certain conditions regarding the reading difficulty. These conditions can be modeled through the objective function and constraints.

In the next section we will discuss relevant work. Section 3 introduces the different parts of the method. In section 4 we evaluate the two main components of the system (lexical and syntactic simplification), and also evaluate how well it is able to reduce the reading difficulty. We end with conclusions in section 5 and indications for future work in section 6.

2. RELATED WORK

2.1 Automatic text simplification

Relevant work started with [4], where sentences were split into shorter sentences by using suppertagging (a weak form of parsing), in order to speed up parsers and improve accuracy. Research in this direction continued with [26], making use of shallow preprocessing and taking hints from punctuation. In this work was also attention for the regeneration stage, so the sentences that were split form a coherent piece of text.

Simplification in order to make text more accessible for aphasic readers was done in [3], in the PSET project. Long sentences and passive constructions are hard to understand for people with aphasia, and these phenomena were simplified making use of the output of a parser and a set of rules thereon. Anaphoric expressions were replaced by there antecedents. The PSET project also had attention for lexical simplification, by replacing difficult words with ones that are easier to understand [8]. The method was evaluated on news

SIGIR: Workshop on Accessible Search Systems 23 July 2010 Geneva, Switzerland

articles from a local newspaper. Similarly, the PorSimples project tries to automatically simplify Brazilian Portuguese text for people with low literacy skills [2]. A first step is classifying each sentence as easy enough or too difficult, using many of the features common in predicting the readability of text. If a sentence is classified as too difficult, a rule based system tries to simplify the sentence. In contrast, the approach we present in this paper decides which sentences to simplify on the level of the entire document, instead of on a per sentence basis.

In a more general setting, the method in [9] can rewrite text by using a Synchronous Tree Adjoining Grammar and large set of paraphrases on this grammar, defined as tree transformations. A global constrained solution is found with Integer Linear Programming (ILP), although only one transformation can be applied to every sentence.

[17] introduced a method to simplify sentences for information seeking applications by extracting Easy Access Sentences. These can loosely be defined as grammatical sentences with one finite verb. Although it has attention for semantically problematic environments, such as conditional constructions, this method generates rather dull texts that are suited for information seeking applications, but not for children. The method in [27] is based on the reduction of parse trees, by applying a set of 242 rules on them, in order to obtain a less 'noisy' dataset for Semantic Role Labeling. However, much information is lost in this process (e.g. modal verbs), and the meaning of the sentences is likely to be altered. The method in [16] serves as a preprocessing step for Relation Extraction, and is based on the Link Grammar parser. Longer chunks of the sentence are fed to the parser until an 'S-link' is found, meaning that that part of the sentence forms a sentence by itself.

2.2 Lexical simplification

Lexical simplification has been performed in [8]. The simplification there consists of lexicon substitution. All words are looked up in WordNet [10] and their synonyms (synsets) are retrieved. For all the synonyms, including the original word, the method looks up the Kucera-Francis frequency [18] in a psycholinguistic dictionary [25]. If one of the synonyms has a higher Kucera-Francis frequency, it is an indication that this synonym is easier, as more frequent words are better known than less frequent ones. If one of the synonyms has a higher frequency than the original, the latter is replaced by the most frequent synonym.

However, in [19] the authors reported that this method often generated "weird sounding" sentences. A possible explanation is that every word can have different unrelated synonyms, because a word can have different meanings. In [8], the argument against using Word Sense Disambiguation (WSD) was that a difficult word might have only one specialized meaning.

2.3 Reading level assessment

Several researchers have demonstrated the effectiveness of machine learning approaches over the traditional measures such as the Flesch-Kincaid and Dale-Chall readability tests. These traditional measures usually are a linear combination of the average sentence length, average number of syllables, and the number of 'difficult' words, e.g. words with 3+ syllables or words that are not in a basic word list. The more advanced features used nowadays range from lexical [5] to syntactic [23, 13], and even coherence [1]. Also cognitively motivated features [11] and discourse structure [24] have been used.

An interesting difference between all these methods is the target audience. [3] focuses on patients with aphasia, although no explicit attempt at identifying the difficult parts of text was made. [22] aims at foreign and second language learners. [11] focuses on people with a cognitive disability. These last two have made use of data obtained from Weekly Reader[1], a magazine with an edition aimed at children of different grades, and thus ideal to train and test reading level assessment approaches.

3. METHOD

Our method consists of three components. The first two are the lexical and syntactic simplification of text. The third component concerns choosing the right set of simplifications that were generated by the previous components.

3.1 Lexical simplification

In the lexical simplification step the aim is to replace difficult words and expressions with simpler ones. This task is closely related to paraphrasing and machine translation, with as source language English, and as target language 'simple' English. Unfortunately, whereas there are parallel corpora available for paraphrasing and machine translation, a similar parallel corpus to learn simplifying expressions from is not available. For this reason we focus our attention on an easier task, the lexical substitution of individual words.

As mentioned in section 2.2, using the most frequent synonyms does not always generate the correct substitutions. Our approach uses a limited form of Word Sense Disambiguation to alleviate this problem. The main idea is that we not only generate alternative words from WordNet, but combine this with a language model [7]. The Latent Words Language model models both language in terms of consecutive words and the contextual meaning of the words as latent variables in a Bayesian network. In a training phase the model learns for every word a probabilistic set of synonyms and related words (i.e. the latent words) from a large, unlabeled training corpus. So rather than taking simply the synonyms from WordNet, we take the intersection with the words generated by the language model (see figure 1 for a graphical representation). Because of the one sense per context phenomenon [28], this gives reasonable grounds to assume the substitutions are correct.

Alternatively, another approach could be to use a standard trigram language model, and ignore the synonyms that have a language model probability below a certain threshold.

What remains is the problem of ranking the different candidates in the intersection of WordNet and the language model, in order to select the easiest. An indication of how easy a word is, could be obtained by looking at the *Age of Acquisition* rating, available from the Oxford psycholinguistic database [25]. Unfortunately, many words lack this rating, so like in previous work we use the Kucera-Francis frequency. The word with the highest frequency is chosen to replace the original word, if it has a higher frequency than the original word.

[1]http://www.weeklyreader.com

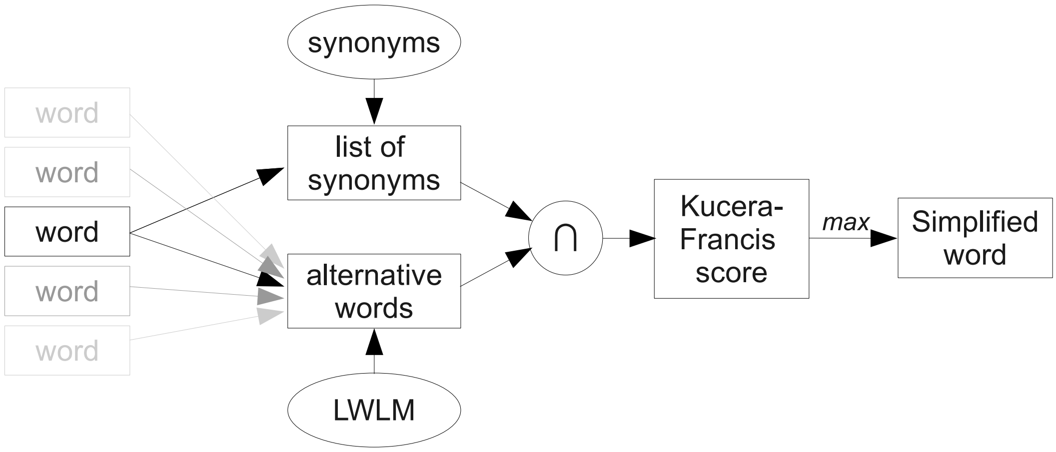

Figure 1: Schematic representation of the lexical simplification

3.2 Syntactic simplification

Previous work has relied on rule based systems to simplify a certain number of syntactic constructions. This is also the approach we follow in this paper. Constructions that are typically simplified are relative clauses, appositions, passive voice, and conjunctions [3], but also constructions such as subordinate clauses and if-then structures [26], which are inspired by Rhetorical Structure Theory (RST).

We use the Stanford parser [6] to perform a syntactic analysis of the input sentences. It has a rich annotation scheme that marks several structures that we aim to simplify. We selected the following set of operations to simplify the sentences:[2]

- Appositions: when an apposition is encountered, it is converted into a new sentence, by introducing an auxiliary verb. The clause it is attached to is copied and made the subject of the new sentence.
  Example: John Smith, a New York taxi driver, won the lottery.
  Becomes: John Smith is a New York taxi driver. John Smith won the lottery.

- Relative clauses: the wh-word is replaced with the word it refers to, and the clause is turned into a new sentence.
  Example: The mayor, who recently got a divorce, is getting married again.
  Becomes: The mayor recently got a divorce. The mayor is getting married again.

- Prefix subordination: this simplification also involves the introduction of new words, slightly based on RST.
  Example: Although it is raining, the sun is shining.
  Becomes: It is raining. But the sun is shining.

- Infix coordination and subordination: trivially, two parts of a sentence connected by 'and' are split into two sentences. If the subject of the first sentence is also the subject of the second, the Stanford parser detects this, and the subject is duplicated. Next to *and*, two sentences conjoined by words such as *although*, *but*, *because*, . . . are also split.

[2]Those that we did not choose to simplify did not occur in the data (if-then constructions), or did not have a significant effect on the readability measure used in the evaluation (activation of passive voice.)

If a sentence can be simplified, and is split into two sentences, then we try to apply the rules again to both of the new sentences. We maintain a list of all possible combinations of rules that can be applied. Thus in this phase, we simply generate all possible simplifications of every input sentence. The actual decision of which rules to apply to which sentences is made by the method described in the next section.

3.3 Optimizing the choice of simplifications

Before starting the section on the Integer Linear Programming formulation, we will first motivate our choice of variable to optimize in order to make the text fit for a reader of a certain age. Afterwards we will extend this to a more general scenario, to incorporate more features.

Numerous features have been used in assessing the difficulty of text. One that recurs many times is the average sentence length. This feature has often been used in the traditional readability measures; the easiness with which it could be calculated probably played an important role. Still, in current research average sentence length is an important feature when training a classifier for readability assessment. It must be noted that in [24], the average sentence length feature is not significantly correlated with the readability, whereas in [11] this feature was found to be significantly different between original and simplified texts.

3.3.1 Integer Linear Programming

A Linear Programming problem consists of decision variables and an objective function, that is a linear combination of the decision variables. Solving the problem means finding an assignment for these variables, so that the objective function is maximized (or minimized). The decision variables can be bounded by linear constraints. In the case of Integer Linear Programming, the decision variables are also constrained to take only integer values. ILP has often been used to find a global solution, for example for dependency parsing [20] and multi-document summarization [12]. One of the first applications of ILP in Natural Language Processing was in the work of [9], whose goal is somehow similar to ours. His goal was to apply paraphrases to sentences in a text, so that the text as a whole conforms to a set of guidelines (e.g., a conference paper that can be no longer than 8 pages). The paraphrases are defined over a Synchronous Tree Adjoining Grammar (STAG). Each paraphrase has a *cost* to apply, and the goal is to make the text conform to the guidelines with a minimal cost. In contrast, in our research the objective function serves to make the text fit a certain age as good as possible. We also take it a step further, by seeing how this is related to research in readability assessment.

3.3.2 Finding a global solution

At the end of the previous step (see section 3.2), we have for every sentence a list of alternative formulations, that can replace the original sentence. For each of these alternatives, we can calculate the influence this will have on the text as a whole. Focussing on the average sentence length, the relevant features that will be influenced by each alternative are the number of sentences and the number of words.

Suppose the original text has S sentences and W words, and sentence $i, \forall i = 1 \dots S$ has n_i possible alternatives, indicated by $a_{i1} \dots a_{in_i}$, and a_{i0} the original sentence[3]. The a_{ij} variables can only be zero or one (a value of one meaning the corresponding alternative should be used), and for a fixed i exactly one of the a_{ij} variables must be one (there can only be one alternative chosen). We can calculate for each a_{ij} the influence this will have on the average sentence length, by calculating the difference in number of sentences, Δs_{ij}, and the difference in number of words, Δw_{ij}, compared to the original sentence. To illustrate with the example of the first rule in section 3.2: the application of this rule ($a_{10} = 0, a_{11} = 1$) would result in an increase of 1 in the number of sentences ($\Delta s_{11} = +1$), and an increase in number of words by 3 ($\Delta s_{11} = +3$).

Stating that the average sentence length should be at most m words per sentence can then be written with the formula:

$$\frac{W + \sum_{ij} a_{ij}\Delta w_{ij}}{S + \sum_{ij} a_{ij}\Delta s_{ij}} \leq m \qquad (1)$$

By rearranging, this equation can be rewritten to the following form:

$$\sum_{ij}(\Delta w_{ij} - m\Delta s_{ij})a_{ij} \leq Sm - W \qquad (2)$$

With the following constraints:

$$a_{ij} \in \{0, 1\} \qquad (3)$$

$$\sum_{j=0}^{n_i} a_{ij} = 1, \forall i \qquad (4)$$

The left hand side of equation 2 can be minimized by using it as the objective function in the ILP formulation, with the constraints from equations 3 and 4. Defining a lower bound on the average sentence length can be done trivially by using equation 2 with a $\geq$ sign instead of the $\leq$ sign, in the form of another constraint. This way the average sentence length isn't made too small, and the text overly simple.

[3]Note that a_{ij} can consist of more than one sentence for $j > 0$.

3.3.3 Extension to more general features

A limitation of this method is that it is not possible to minimize a linear combination of averages, what would be needed for optimization towards e.g. the Flesh-Kincaid score. Because of the two averages in this formula (average sentence length and average syllables per word), the optimization problem becomes a Quadratic Programming problem, which is harder to solve.[4]

It is possible to optimize towards features that are not averages. For example, suppose that we can measure the difficulty of a text by a linear combination of the total number of sentences and the total number of words:

$$\text{difficulty} = \alpha W + \beta S$$

We can then use a similar ILP formulation as in equation 1, so that the difficulty can be minimized by choosing optimal assignments for the variables a_{ij}:

$$\alpha(W + \sum_{ij} \Delta w_{ij}a_{ij}) + \beta(S + \sum_{ij} \Delta s_{ij}a_{ij}) \leq \text{difficulty}$$

Which can be rewritten to:

$$\sum_{ij}(\alpha\Delta w_{ij} + \beta\Delta s_{ij})a_{ij} \leq \text{difficulty} - \alpha W - \beta S$$

with α and β the model parameters, originating from, for example, a linear regression model. Linear regression has been used often in predicting the reading difficulty (e.g. [11, 14]). As long as the features are defined as a total, rather than an average, it is possible to write this in the ILP formulation, and optimize for a certain difficulty. Also the statistical language modeling approach from [5] can be formulated in this way.

In the case that averages are still needed, an alternative solution would be to define upper and lower bounds on each of these features separately, e.g. by taking the average $\mu \pm$ the standard deviation σ, estimated from training data. If the resulting ILP is infeasible, i.e. it is impossible to solve, then the constraints can iteratively be relaxed to fall between $\mu \pm \gamma\sigma$, with $\gamma \geq 1$, until the ILP problem becomes feasible.

4. EVALUATION

4.1 Data

A problem with simplifying text and assessing the reading difficulty of text, is that there is no standard dataset. Because the intended audience is often different (children, students learning a foreign language, people with intellectual disabilities, ...), or the data is protected by copyrights, finding a suitable dataset is not easy. Furthermore, for future research on the simplification of text, it would be convenient if there is a dataset that consists of an original version and a simplified version, so that the latter can be used as a gold standard.

With these objectives in mind, we used data from two publicly available sources, from two different domains. The first part comes from Wikipedia articles. We use the abstracts of the articles on the list of "100 articles every Wikipedia should have". 50 were randomly chosen for the evaluation, the remainder was used for development. Simpler versions of the articles can be found on Simple Wikipedia[5], although

[4]See [9] for details.

[5]http://simple.wikipedia.org

corpus	baseline (synonyms)	our method (+lang. model)
Wikipedia	53.2%	65.0%
Literacyworks	45.9%	57.6%

Table 1: Results of the lexical simplification in terms of precision

the similarity between both versions of the same article is rather low.

The second part of the data comes from the Literacyworks website[6]. It contains news articles from CNN, and every article is accompanied by an abridged version. The abridged version is a simplified form of the original, which is easier to read for students and people that learn English. We randomly selected 50 articles from this set for evaluation.

So in total we have 100 articles, from two different domains.

4.2 Lexical simplification

For the evaluation of the lexical simplification, we randomly selected 180 simplifications from each domain. As a baseline, we compare with the simpler method from [8], discussed in section 2.2. In short, by using a language model we add a weak form of Word Sense Disambiguation to the baseline method, which consists of only selecting the most frequent synonym given by WordNet. The language model was trained on the Reuters corpus.

The evaluation was done using Amazon's Mechanical Turk. Each lexical substitution was graded by three persons, who were asked to indicate whether the substitution was correct or not. The majority vote was taken as the correct answer.

4.2.1 Discussion

It is clear from the results in table 1 that our method, in the third column, outperforms the baseline, shown in the second column. The latter is often too eager to replace words, where our method also looks at the context and makes less errors. This can be illustrated with the following example:

1. Authorities employ (use) various mechanisms to regulate certain behaviors in general.
2. In 2007, about one third of the world 's workers were employed (used) in agriculture.

In sentence 1, both methods replace the word *employ* by the word *use*, which is correct. But in sentence 2, the word *employ* is used in a different context, and the baseline method still replaces it, whereas our method does not.

Table 1 only shows the precision. Empirically, we noticed that the recall is rather low: the most difficult words in the texts are often not replaced. An explanation for this could be that the most difficult words don't have synonyms that are easier to understand. To give a clearer view on this matter, we decided to check how many words are replaceable by a 'simple' word. We started with a list of 3836 unique simple words: the union of the 3000 basic words from the Dale-Chall readability measure and the list of Basic English words that Simple Wikipedia recommends using[7]. For each word, we used WordNet to retrieve the synonyms, thereby ignoring the retrieved synonyms that were already on the initial list of simple words. The total number of unique synonyms was a surprisingly low 10864. Thus, simplifying a text so that it consists entirely out of words from the list of 3836 simple words, is only possible when the input is already limited to the list of 10864 words. Words not in this latter list will not have a synonym, and can not be simplified to a word in the list of simple words. A solution would be to insert elaborations in the text, that explain the meaning of these words, or to leave out the difficult parts by using summarization techniques.

Finally, this experiment only gives an indication on the validity the substitutions, and not of the simplification. Evaluating the latter would require a more extensive evaluation, with children as test subjects (see [15] for example).

[6]http://literacynet.org/cnnsf/

[7]http://simple.wikipedia.org/wiki/Wikipedia:Basic_English_combined_wordlist

Operation	Wikipedia		Literacyworks	
Appositions	23/39	58.9%	35/72	48.6%
Relative clauses	12/20	60.0%	15/35	42.8%
Prefix subordination	2/3	0%	0/0	/
Infix coordination and subordination	30/43	69.7%	78/112	69.6%
Total	67/105	63.8%	128/219	58.5%

Table 2: Accuracy of the syntactic simplification (number correct / number that matched the rule)

Property	Wikipedia	Literacyworks
Nb. of articles	50	50
Nb. of sentences	552	1219
Nb. simplifiable	105	219
Percentage simplifiable	19.0%	18.0%

Table 3: Statistics of the used text data

4.3 Syntactic simplification

We used the same 100 articles from the lexical simplification experiment. We also evaluated the system with Amazon's Mechanical Turk, asking the judges to indicate if the two resulting sentences[8] were still correct English. Again, we used the majority vote out of 3 opinions. To keep the answers simple, we only worked with a binary choice: correct or not correct. The results of the syntactic simplification are in table 2, and details about the data sets are in table 3. The average pairwise inter-annotator agreement was measured with the kappa statistic, and amounts to 0.7, which is reasonable to draw conclusions from.

4.3.1 Discussion

From the results in table 2 it is clear that many errors are made. A lot of the syntactic constructions that we want to simplify are also difficult to recognize for parsers. The task for the parser is made extra hard, because usually long sentences need to be simplified. A lot of the problems come from detecting the boundaries, e.g. finding the clauses that are connected by *and* or finding the end of appositions. The Stanford parser also has problems with lists, separated by commas, as in "I went to Spain, Italy, and Switzerland", in which Italy would be marked as an apposition of Spain.

[8]In most cases, if a sentence could be simplified, it was by only one rule. See section 4.4 for more details.

Property	Wikipedia	Literacyworks
original avg. sentence length	21.6	17.3
minimal avg. sentence length	18.0	14.6
original Flesch-Kincaid grade level	16.2	10.8
minimal Flesch-Kincaid grade level	14.1	9.3

Table 4: The lower bound on average sentence length (words per sentence), and the Flesch-Kincaid grade level. Averaged over 50 articles per type of text

These results might be an indication that the original idea behind text simplification, as a preprocessing step *before* parsing [4, 26], could be worth revisiting. But, since a sentence does not have to be simplified, an easier solution is to analyze it with different parsers, and leave it intact if the difference between the output of the parsers is too large.

4.4 ILP evaluation

To investigate to what end we can simplify the text for a given age, we first let the ILP model make a text that is as simple as possible, by minimizing the average sentence length. These results can be found in table 4, showing the original average sentence length, and the average sentence length after the simplifications. It is clear that the average sentence length is still very high, especially for the Wikipedia articles. The results on the Literacyworks data are better, but still not good enough for the younger children.

When we also include the lexical simplification, we can calculate the Flesch-Kincaid grade level. This is defined as:

$$0.39\frac{\#words}{\#sentences} + 11.8\frac{\#syllables}{\#words} - 15.59$$

The result is a grade level, based on the U.S. education system. Grade 8 corresponds to age 13-14. As can be seen in table 4, the new Wikipedia articles are still far away from a level that is suited for children. The simplified news articles from Literacyworks come closer to the 8th grade, but are still not quite simple enough. That is why we will not perform further evaluation of the global result at this moment, but first put more research into the simplification operations.

For completeness, in figure 2 is a histogram representation of the number of choices that the ILP solver has for each sentence. It is clear that in most cases only one simplification operation can be applied, giving a choice between using the original sentence, or the simplified version. Sometimes an absurdly high number of alternative sentences are generated, the reason for which lies in the interpreting of comma separated lists as appositions, as discussed in section 4.3.1.

5. CONCLUSION

In this paper we have presented a set of methods to simplify text, and simplify the text so that it should better fit the age of the child reading the text. We thereby make an attempt to close the gap between predicting the difficulty of text, and the actual simplification. We improved the accuracy of the lexical simplification with a 11.7% absolute increase, by using a language model to perform a weak form of Word Sense Disambiguation. We implemented a system to split sentences based on the syntax. We relied on the out-

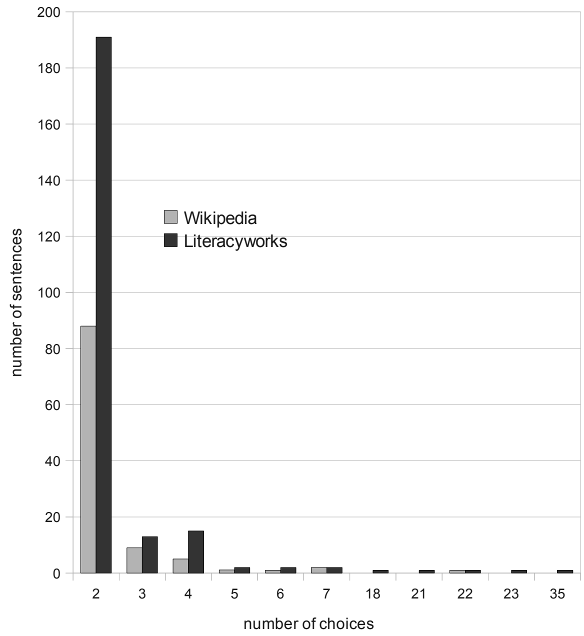

Figure 2: Histogram representation of the number of choices for each input sentence

put of a dependency parser, but it made several mistakes. It appears the constructions we want to simplify to make text more readable for children, are also difficult to understand by parsers.

On a document level, we used an Integer Linear Programming approach to find an optimal choice of simplification operations to perform on the text. We can constrain the ILP formulation to let the features of the text fall between certain boundaries, specific for the age or reading skills of the reader. Unfortunately, with the set of simplification operations we used, it was not possible to reduce the reading difficulty enough for children, at least not without removing information from the text. Simplification of the abstracts of Wikipedia articles resulted in an average decrease of around 2 grade levels according to the Flesch-Kincaid grade level formula, simplification of news articles resulted in a 1.5 grade level decrease. The lexical simplification was unable to simplify the most difficult words, mostly because there is no simple synonym for them. These result show that there are still a lot of possibilities in the field of text simplification.

6. FUTURE WORK

Ignoring the errors that were introduced in the simplification process, existing techniques to simplify text are inadequate to reach a level that is suitable for children. There are however other techniques that can be used as well.

The easiest solution is to incorporate summarization. The disadvantage is that information will be lost from the original text. But in return, it is possible to remove (parts of) sentences that contain difficult words, and to make sentences shorter to decrease e.g. the average sentence length. Care must be taken not to remove parts of the sentence that are being referred to elsewhere in the document, but with the necessary preprocessing steps this can easily be incorporated in the ILP formulation, in the form of additional constraints.

There are also ways to make the text easier, without discarding information. It is possible to introduce elaborations for difficult words. Another option is to do the inverse of a technique in multi-document summarization, i.e. sentence fusion. In sentence fusion, two partly overlapping sentences from different documents are merged together, to create a new sentence that contains the information from both of the sentences. The inverse can be done to make text easier: split a sentence to create partly overlapping sentences, with each a part of the information (e.g. each new sentence has the same subject and verb, but different prepositional phrases).

Ideally, the method should also be able to let the children learn. Rather than making text more understandable, and not let the children learn new words, it would be better from an educational point of view if we could automatically transform the text around a difficult word so that the meaning becomes clear from the context, and the children learn something new.

Finally, we aim to further evaluate these methods, also on Web texts, and see how they can be used in an interactive setting, in order to make the Internet more accessible for children.

Acknowledgments

The research leading to these results has received funding from the European Community's Seventh Framework Programme FP7/2007-2013 under grant agreement n° 231507. We would also like to thank the anonymous reviewers for their valuable comments.

7. REFERENCES

[1] R. Barzilay and M. Lapata. Modeling local coherence: An entity-based approach. *Computational Linguistics*, 34(1):1–34, 2008.

[2] A. Candido Jr, E. Maziero, C. Gasperin, T. Pardo, L. Specia, and S. Aluisio. Supporting the adaptation of texts for poor literacy readers: a text simplification editor for Brazilian Portuguese. In *Proceedings of the Fourth Workshop on Innovative Use of NLP for Building Educational Applications*, pages 34–42. Association for Computational Linguistics, 2009.

[3] J. Carroll, G. Minnen, Y. Canning, S. Devlin, and J. Tait. Practical simplification of English newspaper text to assist aphasic readers. In *Proceedings of the AAAI-98 Workshop on Integrating Artificial Intelligence and Assistive Technology*, pages 7–10. Citeseer, 1998.

[4] R. Chandrasekar and B. Srinivas. Automatic induction of rules for text simplification. *Knowledge Based Systems*, 10(3):183–190, 1997.

[5] K. Collins-Thompson and J. Callan. Predicting reading difficulty with statistical language models. *Journal of the American Society for Information Science and Technology*, 56(13):1448–1462, 2005.

[6] M.-C. de Marneffe, B. MacCartney, and C. D. Manning. Generating typed dependency parses from phrase structure parses. In *In Proceedings of LREC-06*, pages 449–454, 2006.

[7] K. Deschacht and M.-F. Moens. The Latent Words Language Model. In *Proceedings of the 18th Annual Belgian-Dutch Conference on Machine Learning*, 2009.

[8] S. Devlin and J. Tait. The use of a psycholinguistic database in the simplification of text for aphasic readers. *Linguistic Databases*, pages 161–173, 1998.

[9] M. Dras. *Tree adjoining grammar and the reluctant paraphrasing of text.* PhD thesis, Macquarie University NSW 2109 Australia, 1999.

[10] C. Fellbaum. *WordNet: An electronic lexical database.* MIT press Cambridge, MA, 1998.

[11] L. Feng, N. Elhadad, and M. Huenerfauth. Cognitively motivated features for readability assessment. In *Proceedings of the 12th Conference of the European Chapter of the Association for Computational Linguistics (EACL)*, pages 229–237. Association for Computational Linguistics, 2009.

[12] D. Gillick and B. Favre. A scalable global model for summarization. In *Proceedings of the Workshop on Integer Linear Programming for Natural Langauge Processing*, pages 10–18. Association for Computational Linguistics, 2009.

[13] M. Heilman, K. Collins-Thompson, J. Callan, and M. Eskenazi. Combining lexical and grammatical features to improve readability measures for first and second language texts. In *Proceedings of NAACL HLT*, pages 460–467, 2007.

[14] M. Heilman, K. Collins-Thompson, and M. Eskenazi. An analysis of statistical models and features for reading difficulty prediction. In *Proceedings of the Third Workshop on Innovative Use of NLP for Building Educational Applications*, pages 71–79. Association for Computational Linguistics, 2008.

[15] M. Huenerfauth, L. Feng, and N. Elhadad. Comparing evaluation techniques for text readability software for adults with intellectual disabilities. In *Proceeding of the Eleventh International ACM SIGACCESS Conference on Computers and Accessibility*, pages 3–10. ACM, 2009.

[16] S. Jonnalagadda, L. Tari, J. Hakenberg, C. Baral, and G. Gonzalez. Towards effective sentence simplification for automatic processing of biomedical text. In *Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics, Companion Volume: Short Papers*, pages 177–180. Association for Computational Linguistics, 2009.

[17] B. Klebanov, K. Knight, and D. Marcu. Text simplification for information-seeking applications. *On the Move to Meaningful Internet Systems, Lecture Notes in Computer Science*, pages 735–747, 2004.

[18] H. Kučera and W. Francis. *Computational analysis of present-day American English.* University Press of New England, 1967.

[19] P. Lal and S. Ruger. Extract-based summarization with simplification. In *DUC 2002: Workshop on Text Summarization, July 11–12, 2002, Philadelphia, PA, USA*, 2002.

[20] A. Martins, N. Smith, and E. Xing. Concise integer linear programming formulations for dependency parsing. In *Proceedings of Annual Meeting of the Association for Computational Linguistics (ACLIJCNLP 09), Singapore*, 2009.

[21] F. Oliveira, F. Wong, and I. Hong. Systematic

processing of long sentences in rule based Portuguese-Chinese machine translation. *Computational Linguistics and Intelligent Text Processing*, pages 417–426, 2010.

[22] S. Petersen. *Natural language processing tools for reading level assessment and text simplification for bilingual education*. PhD thesis, University of Washington, 2007.

[23] S. Petersen and M. Ostendorf. A machine learning approach to reading level assessment. *Computer Speech & Language*, 23(1):89–106, 2009.

[24] E. Pitler and A. Nenkova. Revisiting readability: A unified framework for predicting text quality. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 186–195. Association for Computational Linguistics, 2008.

[25] P. Quinlan. *The Oxford psycholinguistic database*. Oxford University Press Oxford, 1992.

[26] A. Siddharthan. Syntactic simplification and text cohesion. *Research on Language & Computation*, 4(1):77–109, 2006.

[27] D. Vickrey and D. Koller. Sentence simplification for semantic role labeling. In *Proceedings of ACL-08: HLT*, pages 344–352, Columbus, Ohio, June 2008. Association for Computational Linguistics.

[28] D. Yarowsky. One sense per collocation. In *Proceedings of the Workshop on Human Language Technology*, page 271. Association for Computational Linguistics, 1993.